

REMARKS/ARGUMENTS

Claims 13-24 were pending in this matter and all have been rejected. Claims 13 - 24 have been amended. New claims 25-28 have been added. Applicants respectfully request reconsideration of the application for the following reasons.

Claim 13 has been amended to specify that monitoring the at least partial run of the program is to capture information on interaction between each pair of components, wherein the information on interaction comprises information relating to an explicit request from one component of the pair to the other component of the pair for some data, and wherein monitoring comprises recording the source component, the destination component and the amount of data. The preamble has been amended to remove the language of “the properties comprising string representation and data structure.” Moreover, step d) has been amended to read: “determining a choice set of interrelated choices of the data representation properties of each component which minimizes total cost of the at least partial run.” Support for the amendments is found at Page 2 Para 24 of the detailed description. Thus “*operation* in which data flows from one variable to another” is inherent in that it is *direct* flow of information (not *indirect*, e.g. where one instruction happens to output data into a register and some other instructions happens to input that data from the register), as well as *explicit reference/invocation* i.e. one component actually calls the other to get some data, as opposed to *coincidence* i.e. two instructions just happen to end up in flight together, as a matter of luck.

CLAIM REJECTIONS UNDER 35 U.S.C. §102

The Office Action rejected claims 13, 14, 16, 19, 20, and 22 as being anticipated by US Patent No. 6,360,360 to Bates (hereafter "Bates"). Applicant respectfully submits that the claims as amended are not anticipated by Bates.

With respect to claim 13, Bates discloses simply “a function call profile is constructed by determining the set of functions in an object that are called within the program and estimating the number of times each is called.” Bates does not capture, or use in analysis, the source of a call—

the *caller*; instead Bates uses only the destination of the call—the *callee*, whereas claim 13 requires capturing and recording *both* source and destination, as well as amount of data passed. Accordingly, Bates can only perform a local optimization whereas the claimed invention performs a global optimization.

Bates discloses “selecting an implementation” for an object (which is equivalent to setting an implementation property of the object). Essentially, Bates looks at how much each function of an object is called, and then he selects an implementation for the object in which the most frequently called functions are relatively cheap – Bates selects the implementation with the lowest weighted cost (that cost being constructed by summing over all functions the product of the number of calls of the function times the cost of that function in that implementation). Bates does a local optimization – Bates will choose whatever’s best for that object. Bates does that in isolation from the choices made for any other object. But that might not be best for the program as a whole, because the conversion required in passing the data from the callee to the caller might be so expensive that it would have been better to select a higher cost implementation of the callee that would incur less conversion cost in passing the data to the caller.

Instead, the claimed invention makes a set of choices, one for each component in a group, with all of the choices for that group being interrelated. Taking into account the caller, the representation used by the caller, and the cost of converting between the representation of the callee and that of the caller. The claims at issue require a global optimization. For example, consider the interrelated choices minimize *total* cost. Bates teaches optimizing for each object, in contrast to the claimed invention which optimizes for the entire group. This global optimization may select a representation that makes method execution take longer for a component (i.e. an implementation with higher weighted cost), but which will result in lower conversion costs as data is passed from one component to the other, thereby resulting in lower overall cost for the group as whole.

Therefore, Bates neither teaches nor suggests the limitation of step d) because the determination of choices of step d) minimizes *total* cost of the at least partial run.

Claims 14-18 are dependent on claim 13 and hence are not anticipated for at least the foregoing reasons.

Claim 19 is a computer program product counterpart of claim 13 and is not anticipated for at least the foregoing reasons. Claims 20-24 are dependent on claim 19 and hence are not anticipated for at least the foregoing reasons.

The Office Action rejected claims 13, 14, 16, 19 20, and 22 as being anticipated by US Patent No. 6,092,180 to Anderson (hereafter "Anderson"). Applicant respectfully submits that the claims as amended are not anticipated by Anderson.

Anderson uses "interaction" in the same sense as in drugs (i.e. coincidental concurrency leading to a bad effect e.g. if you take two drugs together, you develop breathing problems, but it is unknown why (so you don't take the two drugs together anymore), or e.g. if two instructions are in flight together, one of them stalls, but who knows why (we don't try to figure that out, we just schedule so that they're not in flight together in the future)). This teaches away from claim 13 which uses "interaction" in the sense of an explicit request from one component to another, along with transfer of data from one to the other.

The kinds of interactions addressed by the claims – requests from one component to another – are the essence of the program – its logic. The interactions will always occur no matter how the data representation properties are set. Our goal is to lower the cost of these interactions by setting the data representation properties appropriately.

The kinds of interactions Anderson addresses – latencies and bottlenecks due to resource conflicts and other dependencies between instructions that happen to be in flight together – are not inherently the essence of the program, but rather are unfortunate occurrences that result from instructions interfering with each other. Anderson's goal is to eliminate as many of these as possible by changing the scheduling appropriately.

In the "at least partial run," Anderson only does *sampling* (i.e., does not look at everything, but instead just looks at some randomly selected instructions, and then does statistical inference to try to estimate what happens overall), whereas claim 13 requires

monitoring to determine information on **each** pair of components (i.e. capture information on all interactions among all components).

Anderson changes scheduling (i.e., which “slot” an instruction will be in on the “conveyor belt” feeding instructions into the processor), which is in effect changing an *implementation* property, which clearly is not changing a *data representation* property, whereas the claims as amended change or set a data representation property.

Anderson discloses average frequency of sampling, which is the frequency at which information is captured about the execution of instructions, and it relates to how often information about an instruction is captured. It is a characteristic of the sampling, not of the execution of the program itself. Claim 14 requires capturing information about the frequency of interaction between each pair of interacting components, which is indeed information about the execution of the program. It is not about the monitoring of the program execution, and hence is not analogous to Anderson’s frequency which is about the sampling of the program execution.

Anderson discloses profile registers that do capture many useful facts, but they do not capture explicit requests from an invoker to an invokee and the amount of data passed back and forth, as required by claim 13.

Anderson discloses estimation of stall cycles in some cases, but estimating stall cycles by subtracting required work cycles from fetch-to-retire latency is not analogous to computing the cost to convert a certain amount of data from one representation to another, as required by claim 13.

Anderson discloses defining a window of instructions that may execute concurrently, but that is for purposes of setting up sampling and doing analysis, not for purposes of actually making changes to subsequent executions that will minimize cost, as required by claim 13.

CLAIM REJECTIONS UNDER 35 U.S.C. §103

Claims 15, 17, 21 and 23 were rejected under 35 U.S.C. §103 as being unpatentable over Anderson in view of alleged admitted prior art. Applicant respectfully submits that claims 15, 17, 21 and 23, as amended, are patentable over Anderson in view of the alleged admitted prior art.

Claims 15 and 17 include the limitations added to claim 13 by virtue of their dependence on claim 13. As discussed above, the combination of Anderson and the alleged admitted prior art does not teach or suggest those limitations.

Similarly, claims 21 and 23 are dependent on claim 19 and include the limitations added to claim 19. As shown above those limitations are neither taught nor suggested by the cited combination of references. Further, Chadhuri discloses a *tree* data structure. Claim 19 requires a *graph* data structure. Chadhuri's structure would be inadequate for purposes of the claimed invention.

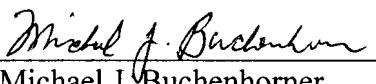
Claims 18 and 24 were rejected under 35 U.S.C. §103 as being unpatentable over Anderson in view of U.S. Patent No. 5,598,559 to Chadhuri. Chadhuri relates to techniques for optimizing database queries and, more particularly, to a method and apparatus for optimizing queries having group-by operators. Chadhuri does not teach modifying systems such as Anderson to perform the method of claim 13 or the medium of claim 19, as those claims have been amended, and claims 18 and 24 are dependent on claims 13 and 19 respectively.

Furthermore, Chadhuri discloses generating a plurality of execution plan tree data structures, and then choosing the execution plan having the lower estimated cost. In contrast, claim 18 requires graph cutting techniques to find a minimum cut of the single graph that we construct. Chadhuri's approach does not address our purposes.

Serial Number 10/073,628
Attorney Docket Number YOR920020022US1
Amendment 091407 Page 12 of 12

For the foregoing reasons, Applicant respectfully requests allowance of the pending claims and that a timely Notice of Allowance be issued in this case.

Respectfully submitted,



Michael J. Buchenhorner
Reg. No. 33,162

Date: September 14, 2007

Michael Buchenhorner P.A.
8540 SW 83 Street, Suite 100
Miami, Florida 33143
Telephone: (305) 273-8007
Facsimile: (305) 595-9579